2019 International Conference on Robotics and Automation (ICRA)
Palais des congres de Montreal, Montreal, Canada, May 20-24, 2019

Optimal Proactive Path Planning for Collaborative Robots in Industrial Contexts

Andrea Casalino[1], Davide Bazzi[1], Andrea Maria Zanchettin[1] and Paolo Rocco[1]

Abstract— **The coexistence of humans and robots in the future production plants is one of the pillars of Industry 4.0. Humans and robots will collaborate to accomplish common tasks in order to mutually compensate their deficiencies. In recent years, many efforts have been spent to develop safe motion planning strategies, designed to prevent robots from injuring humans. Most of the previous techniques are classifiable as reactive, since the considered motion controllers impose some local corrective actions in order to dodge the space occupied by the human. In this paper, a proactive approach is adopted, optimizing robotic paths according to a prediction of the volume occupied by the human when collaborating with the robot. The validity of the approach is shown in a realistic use-case involving the collaboration of a human operator with a 7 degrees robotic arm, the ABB YuMi.**

I. Introduction

In the past years, robots were mainly conceived to work alone in highly structured environments, replacing humans in carrying out activities that were repetitive, dangerous or requiring high precision. In such contexts, robots were physically segregated from humans by means of barriers. Only recently the topic of human-robot collaboration has attracted much attention. Such interest is motivated by the Industry 4.0 paradigm [1], which considers as a fundamental pillar the massive presence of robots in production plants, cooperating with humans. The factories of the future will adapt their behaviours, reacting to rapidly changing production plants. In this scenario, robots can no longer be adopted to simply accomplish repetitive tasks. Instead, humans and robots will both adapt and synchronize in many ways, collaborating to accomplish common tasks.

Clearly, one fundamental requirement for a cooperation is the safe coexistence in a shared space. To this purpose, a new generation of robots, called cobots, were specifically designed to be deployed in collaborative workplaces[1], and many works started addressing the problem of safe motion planning. However, the majority of the developed approaches are classifiable as reactive: the robot is slowed down along its nominal path or it is forced to undertake local dodging manoeuvres, in case imminent collisions are detected. One example is the strategy described in [2], where a repulsive field deforms the trajectory of the robot, in order to let it accomplish its task, but in a safe way for the human. A similar approach was adopted in [3], where the accelerations of the robot are modulated according to a scaling factor, computed by solving a constrained optimization problem. The aim is to modify as little as possible the initial path, to guarantee the satisfaction of a safety constraint. In [4], the previous approach was extended, to tackle the problem with a model predictive control perspective, optimizing the accelerations on a longer time horizon.

The aim of this work is instead to compute proactive trajectories, i.e. trajectories designed to reduce in advance the risk of collision with humans, without waiting for the situation to become critical. To the best of the authors's knowledge, only a few prior works can be found addressing the specific problem of proactive path planning. On the opposite, proactivity was used in different ways, for instance as a tool for clarifying the intention of a human mate. To this purpose, [5] introduced an ensemble of probabilistic state machines for inferring human intention. When the inference process truns out to be partially uncertain, the robot undertakes specific actions, to provoke a reaction in the human mate, with the aim of clarifying the operator's intention. A similar approach is adopted in [6], that exploits Dynamic Bayesian Networks for inference. The only examples of proactive path planners can be found in [7], [8] and [9]. [7] addressed the problem of co-planning, computing the trajectories for both a human and a mobile robot. The proposed approach is however able to plan only 2D planar paths. [8] is another example of a 2D planner, where optimal paths for a mobile cobot are computed considering not only the safety of the human mate but also the acceptance of the resulting motion, trying to avoid trajectories for which the robot goes out from the visibility cone of the human. [9] tackled the proactive planning problem for an articulated robot, even though the developed method relies on an off-line computation of the paths.

The computation of proactive trajectories requires to estimate which human activities are likely to be executed simultaneously with a specific robotic one. This kind of modelling is not required by reactive approaches, which compute paths that are only locally optimal. On the opposite, proactive paths aim to be globally optimal, since the whole path is optimized. The reactive and the proactive approaches can be also combined: a proactive path can be computed minimizing the probability of collisions with the human, but then its execution can be managed by a reactive motion controller, enforcing some additional safety constraints.

The rest of this article is structured as follows. Section II explains how the collaboration between a human and a robot is modelled for the purpose of proactive planning. The computation of a probabilistic description of the volume occupied

[1] The authors are with Politecnico di Milano, Dipartimento di Elettronica, Informazione e Bioingegneria, Piazza L. Da Vinci 32, 20133, Milano, Italy (e-mail: name.surname@polimi.it).

[1]See e.g. http://blog.robotiq.com/collaborative-robot-ebook.

by the human is detailed in Section III and exploited by the algorithm computing the proactive paths, as explained in Section IV. Section V presents the experimental results, while Section VI provides some concluding considerations.

II. Modelling the Human-Robot Collaboration

The aim of this Section is to show how the interaction between a human and a robot is modelled for the aim of proactive path planning. The typical scenario is a collaborative assembly for which a certain number of activities have to be accomplished to compose some products. Some of them are assigned to the robot and others are undertaken by the human. Suppose also that actions are preassigned to agents, according to their capabilities.

Among all actions, we consider those for which the human or the robot have to traverse a shared working area. Sets $\mathcal{H} = \{h_1, \cdots, h_n\}$ and $\mathcal{R} = \{r_1, \cdots, r_m\}$ contain such kind of actions, respectively for the human and the robot.

For every $r_j \in \mathcal{R}$, a corresponding nominal path Q_j is preassigned. Q_j ensures the completion of r_j, but ignoring the presence of humans in the workspace. It is described by a series of intermediate waypoints q_{jk}'s. In terms of joint configurations:

$$Q_j = \begin{bmatrix} q_{jo} & q_{j1} & \cdots & q_{jN-1} & q_{jN} \end{bmatrix} \tag{1}$$

The objective of the proposed approach is to compute for every r_j, a proactive path Qp_j, modifying the waypoints of Q_j, in order to minimize a certain global cost $\mathcal{J}(Qp_j)$ (see also Section IV). The shape of $\mathcal{J}$ is designed to obtain a Qp_j as much as possible similar to Q_j, but for which the probability of a collision with the human is minimized (see again Section IV). On the opposite, a pure reactive approach would be only able to locally deform a trajectory, executing on-line some near range corrective actions, obtaining most of the time a suboptimal solution. An adaptive approach is adopted, periodically recomputing proactive paths, according to the most recent data describing the motion of the operator. To obtain such a description, the trajectories performed by the operator while executing actions in $\mathcal{H}$ (which are basically sequence of postures), are retrieved from a depth camera[2] and segmented on-line. Each acquired realization of a certain h_i is saved as an additional sample in a specific database (see Figure 1).

P_{ji} will be used to indicate the conditional probability that the human is doing a certain action h_i while the robot simultaneously executes a particular action r_j. The function $h(t)$ will describe the actions performed during time by the human. $h(t)$ takes values in a discrete set $\{0, 1, \cdots, n\}$, according to what action in $\mathcal{H}$ the human is doing for the specified time t. A value equal to 0, indicates that the human is not doing any of the actions in $\mathcal{H}$. $r(t)$ is a similar function defined considering actions in $\mathcal{R}$, i.e. those of the robot. The definition of P_{ji} becomes:

$$P_{ji}(t) = \mathbb{P}(h(t) = i | r(t) = j, \quad \forall t) \tag{2}$$

[2]https://developer.microsoft.com/it-it/windows/kinect

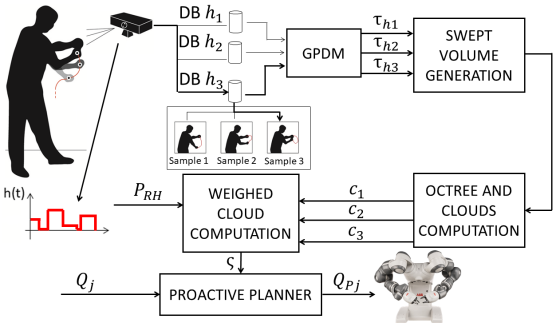

Fig. 1: The entire pipeline of the approach. A depth camera records the motion of the human, whose trajectories are then segmented and stored in different databases, one for each human action. Such samples are considered by GPDM to compute regressed trajectories for the human motion, which in turn are exploited to compute the corresponding probability clouds and the proactive paths.

The computation of $P_{ji}(t)$ relies on $I_r(t)$ and $I_h(t)$, which are two indicator functions defined as follows:

$$I_r(t) = \begin{pmatrix} I_{r1}(t) & \cdots & I_{rm}(t) \end{pmatrix}^T$$
$$\text{such that } I_{rj}(t) = \begin{cases} 1, & \text{if } r(t) = j \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

Such a function indicates whether the robot, at time t, is executing r_j. $I_h(t)$ is similarly defined, considering $h(t)$, hence the human. Every P_{ji} can be computed considering a sliding temporal window of length T:

$$P_{RH}(t) = \begin{pmatrix} P_{11}(t) & \cdots & P_{1n}(t) \\ \vdots & \ddots & \vdots \\ P_{m1}(t) & \cdots & P_{mn}(t) \end{pmatrix} = \\ = M \cdot \int_{t-T}^{t} I_r(\tau) \cdot I_h(\tau)^T d\tau \tag{4}$$

$$M = \begin{pmatrix} \frac{1}{\int_{t-T}^{t} I_{r1}(t)dt} & 0 & \cdots & 0 \\ 0 & \frac{1}{\int_{t-T}^{t} I_{r2}(t)dt} & 0 & \vdots \\ & & \ddots & \\ 0 & \cdots & 0 & \frac{1}{\int_{t-T}^{t} I_{rm}(t)dt} \end{pmatrix} \tag{5}$$

Even when considering a set $\mathcal{H}$ with a large cardinality, distributions in P_{RH} will have a low entropy, if we assume that the human actions follow a certain pattern (which for industrial contexts is an acceptable assumption). Indeed, few human actions would be likely to be executed simultaneously to a specific robot one.

III. Probabilistic Description of the Human Motion

The starting time of a specific action h_j is a random variable having high variability [10]. For this reason, a probabilistic time-independent description of h_j is adopted. At a first stage, a trajectory τ_j is obtained exploiting the samples acquired for h_j. τ_j is then employed to compute a

probabilistic point cloud C_j describing the volume occupied by the human while executing h_j (see the left part of Figure 1). The computation of a single τ is detailed in Section III-A, while the steps involved in the determination of the corresponding C are reported in III-B.

A. Modelling the human trajectories

The human motion is modelled by making use of Gaussian Process Dynamical Model (GPDM) [11]. GPDM are an extension of the canonical Gaussian Processes (GP), which are probabilistic non parametric models [12], adopted to learn static mappings (unknown functions describing input-output correlations). The method developed in [11] will be briefly reviewed in the rest of this Section, as it will be adopted in this work. GPDM is a latent variable model with a non-linear probabilistic mapping from a latent space $\mathcal{X}$ to an observation one $\mathcal{Y}$ and a non-linear dynamics evolving in the latent space. Every element $y \in \mathcal{Y}$ is a collection of information describing the human posture, as for example the position in the space of some points of interest like the elbows, the wrists and others. The training set consists of a collection of acquired trajectories $\{\langle y_1, \cdots, y_{L1}\rangle, \cdots, \langle y_1, \cdots, y_{Lk}\rangle\}$, for which the corresponding latent space sequences $\langle x_1, \cdots, x_{Lj}\rangle_{j=1,\cdots,k}$ are not known, and are therefore treated like additional hyper-parameters to be learnt by the training process, see [11]. The problem amounts essentially to learn two Gaussian Processes simultaneously: the first one approximating a function $f(x)$ which describes the latent dynamics, and the second one approximating $g(x)$, i.e. the mapping between $\mathcal{X}$ and $\mathcal{Y}$. The obtained GPDM, can be used to compute the regressed trajectory $\tau = \langle \tau_1, \cdots, \tau_{L\tau}\rangle$, which models the human motion for a specific action; propagating forward in time an initial x_1, leading to:

$$\tau_k = \mathbb{E}[g(x_k)]; \qquad x_{k+1} = \mathbb{E}[f(x_k)] \tag{6}$$

B. Computing the probability cloud

The computation of τ described in the previous Section is the starting point for the determination of the corresponding probability cloud C, which is part of the contribution of this work. C consists in a set of g points in the space with an associated probability of occupation:

$$C = \{\begin{bmatrix} p_{C1} \\ \gamma_1 \end{bmatrix}, \cdots, \begin{bmatrix} p_{Cg} \\ \gamma_g \end{bmatrix}\}$$
$$p_{C1,\cdots,g} \in \mathbb{R}^3, \;\; \gamma_{1,\cdots,g} \in [0,1] \tag{7}$$

C is computed according to the volume swept by the operator when executing τ. We are not interested in describing the volumes swept by the entire body of the operator, but we can limit our analysis to the motion of the forearms and arms, since those parts are typically the ones that effectively share the space with the robot. Moreover, without loss of generality, it is assumed that every action h_j is performed by moving one single arm.

The volume swept between two intermediate poses τ_k, τ_{k+1} for the forearm or the arm, is assumed to be the Minkowski sum of a sphere[3] and a polytope whose vertices are the positions of the involved skeletal points (wrist and elbow or elbow and shoulder) for the two steps k and $k+1$. Therefore, such volumes have convex shapes, that can be approximated by using OctTrees [13] (see Figure 2). The latter approximation is useful for computing in a faster way the probabilistic cloud C. Indeed, the computation of C is made considering a discretized 3D grid of g equally spaced elements. Such set contains the nodes of the grid obtained by partitioning into G parts every dimension of the smallest oriented bounding box entirely containing the set of volumes swept when performing τ (refer to Figure 2). Clearly, it holds that $G = g^3$.

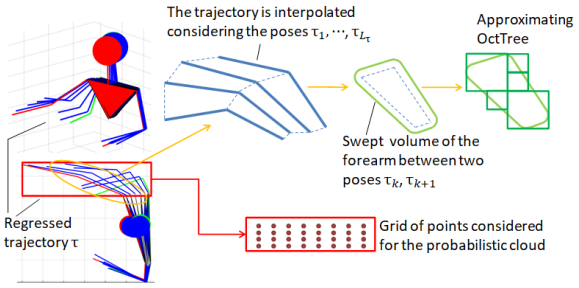

Fig. 2: Pictures on the left depict an example of regressed trajectory taken from two distinct views. The red box bounding the trajectory is considered for defining a grid of g points. The volume swept by the arm of the operator, is approximated by the light green convex set in the middle, which in turn is approximated by the dark green OctTree (only the OctTree of the upper part of the arm is reported).

For every pair τ_k τ_{k+1}, a vector c_k of g elements must be computed. Every element in c_k can be equal to 1 or 0, depending on the circumstance that the corresponding point in the grid is contained or not by the volumes swept from k to $k+1$. The computations are speed up considering the approximating OctTree: the points in the grid contained in every leaf of the tree can be easily extracted and the relative value in the binarized vector c_k is set equal to 1. The occupancy probabilities γ are computed as the mean value of c along the trajectory τ:

$$\gamma = \begin{pmatrix} \gamma_1 \\ \vdots \\ \gamma_g \end{pmatrix} = \frac{1}{L\tau - 1} \sum_{k=1}^{L\tau - 1} c_k \tag{8}$$

where $L\tau$ is the number of poses contained in τ. Those points for which the corresponding γ would result to be 0 or below a certain threshold, can be discarded from the clouds.

IV. Proactive path planning

For every action $h_i \in \mathcal{H}$, a probabilistic cloud C_i is computed by following the steps detailed in the previous Section. When computing the proactive path associated to

[3]The radius of the sphere is chosen by considering the typical anatomical size of human arms.

the generic robot action r_j, a corresponding cloud ς_j must be considered. ς_j is a weighted union of all the clouds $C_{1,\cdots,n}$ describing the volume occupancy for every human actions:

$$\varsigma_j = \{\{P_{j1} \odot C_1\} \cup \cdots \cup \{P_{jn} \odot C_n\}\} \tag{9}$$

where the operator $\odot$ applies as follows:

$$w \odot C = \{\begin{bmatrix} p_{C1} \\ \gamma_1 \cdot w \end{bmatrix}, \cdots, \begin{bmatrix} p_{Cg} \\ \gamma_g \cdot w \end{bmatrix}\} \tag{10}$$

This is done for considering not only the volume occupied by the human when executing specific actions, but also the probability that he/she is doing those actions simultaneously to r_j.

Once ς_j is available, the proactive path Q_{pj} is computed as the result of an optimization problem, whose cost function $\mathcal{J}$ balances the risk of collisions with the human and the need to alter as little as possible Q_j. Figure 1 summarizes the entire pipeline of the approach.

The points in the resulting cloud ς_j, induce a radial repulsive field, for which the potential can be evaluated for a generic point in the space. The value of the potential of ς_j, as explained below, is taken into account by the cost function. $\mathcal{J}$ is a summation of terms, one for every waypoint $q_{1,\cdots,N-1}$, characterizing the path[4]:

$$\mathcal{J} = \sum_{k=1}^{N-1} J(q_k) \tag{11}$$

We propose to adopt for proactive planning, a cost $J(q_k)$ for the generic waypoint obtained as a summation of three terms:

$$\begin{aligned} J(q_k) &= J_{Ob}(q_k) + \lambda_1 \cdot J_{Rep}(q_k) + \lambda_2 \cdot J_{Diff}(q_k) \\ J_{Rep}(q_k) &= \sum_{t \in T} \sum_{p_{Cj} \in \varsigma_j} \gamma_i \exp\left(-\alpha \|p_{Cj} - t\|_2\right) \\ J_{Diff}(q_k) &= \|q_{jk} - q_k\|_2 \end{aligned} \tag{12}$$

$J_{Ob}(q_k)$ is the collision cost: it's equal to the summation of the penetration depths of every robotic links in every fixed obstacles. For convex shapes, $J_{Ob}(q_k)$ can be computed by making use of the GJK algorithm in combination with EPA, refer to [14]. J_{Diff} discourages big deformations of the path w.r.t. the nominal one, acting like an attractive field that counterbalances the repulsion induced by the probability cloud. J_{Rep} it's the value assumed by the potential of a field induced by ς_j, evaluated in a series of points $T = \{t_1, t_2 \cdots\}$ along the kinematic chain of the manipulator. $\lambda_{1,2}$ are tunable parameters balancing the importance of every term in J. Figure 3 summarizes the above considerations. The minimum for $\mathcal{J}$ is found by applying the STOMP algorithm [15], which is a sample based algorithm that iteratively deforms an initial robotic path, for optimizing cost $\mathcal{J}$. This results in a kind of gradient descend applied to path planning. The main advantage with respect to similar strategies, is that STOMP does not require to know in a closed form the gradient $\partial_Q \mathcal{J}$, allowing large flexibility when designing the cost function. Figure 4 reports different results obtained by STOMP, varying the parameters $\lambda_{1,2}$, when considering the nominal path assigned to YuMi for the experiments (see Section V) and the probability cloud reported in the picture at the right corner of Figure 7.

[4] q_o and q_N are not considered here because the starting and the ending poses are fixed.

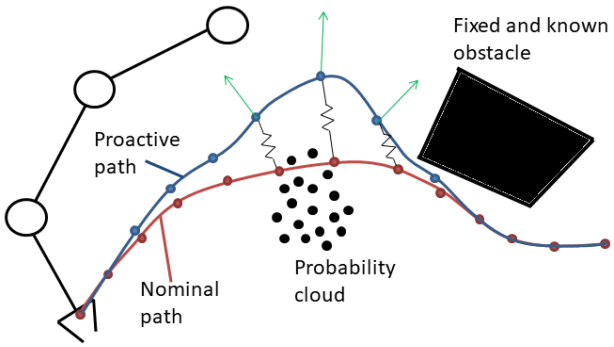

Fig. 3: The initial path of the manipulator (red) and the corresponding proactive one (blue). The probability cloud induces the repulsive field depicted with green arrows. The effect produced by the springs, refers to the term J_{Diff} in the cost function. The black shape on the right upper corner is a fixed known obstacle.

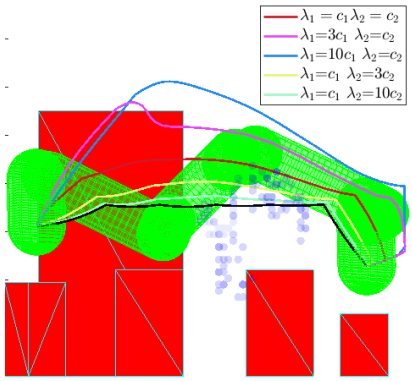

Fig. 4: Different obtained results of STOMP, varying the proportion between the coefficients $\lambda_{1,2}$.

V. Experimental validation

The proposed method has been tested on a realistic human-robot collaborative interaction. The experimental setup consists of the ABB dual-arm robot YuMi, and a Kinect camera, see Figure 5. Both the Kinect and YuMi are connected to a quad core Intel NUC (https://www.intel.com/content/www/us/en/products/boards-kits/nuc.html) , which collects the information from the camera and computes the proactive paths for the robot.

A. Description of the collaborative assembly

The goal is to assemble a box containing a USB pen drive, through the following steps:

- action 1: take one newer box
- action 2: insert in the box one layer of foam and the USB pen drive

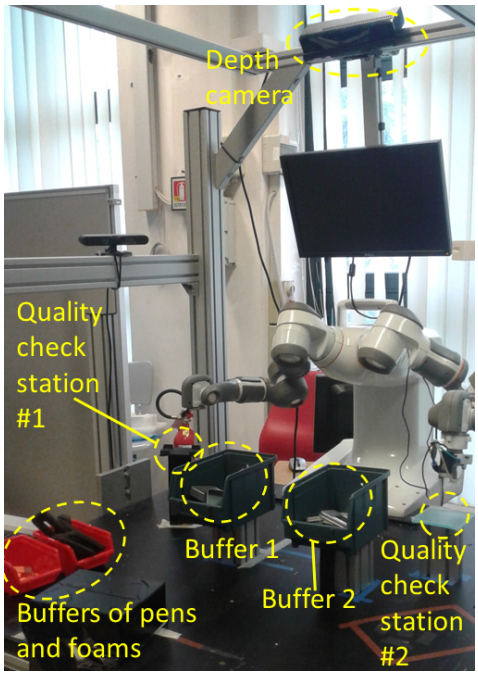

Fig. 5: The experimental setup. The left arm of YuMi is involved in the collaborative assembly.

- action 3: bring the filled box to the first quality check station
- action 4: execute a first quality check
- action 5: take the filled box and put a cover
- action 6: put the assembled box on a second quality check station
- action 7: perform the final quality check

Actions 1,2,3,5 and 6 are executed by the human, while 4 and 7 are assigned to YuMi. Items are stored in buffers, which are located as indicated in Figure 5. During the assembly, it's up to the operator to choose buffer 1 or 2 when executing action 1 or 5. After the human completes action 3, YuMi starts action 4, after which it moves to the second quality check station waiting for the human to execute action 6. $\mathcal{R} = \{r_1\}$. r_1 is the motion of the robot between the two quality control stations. After the second quality check, the robot goes to a position far from the shared space with the human, waiting for a new command. $\mathcal{H}$ is composed of h_1 and h_2. h_1 consists in the picking from buffer 1 (a box or a cover), while h_2 is a similar action for buffer 2. Two different databases store samples for h_1 and h_2. Once a newer sample is available, the oldest one is deleted and the corresponding database is enriched with the newer one (FIFO logic). Every time a new proactive path is computed, the regressed trajectories τ_1, τ_2 are recomputed, considering the samples contained at that time in the buffers. The capacity of the databases was set equal to 3 samples for the proposed experiments. This choice was made to represent the human motion with a restricted number of recent samples, in order to acquire a certain level of adaptation.

B. Experiments

We recruited 14 participants for our experiments, which were divided into two groups of 7 each. The first group was asked to perform the collaborative assembly with the robot performing proactive paths, while for the others the robot has persistently executed its nominal path. For both the groups, the robot's motion was controlled with the strategy described in [16], modulating its speed along a fixed path (respectively a proactive one and a path agnostic of the human presence). When considering the experiments for the first group, the proactive computation loop recomputed P_{11}, P_{12} and a newer proactive path every 20 s, in order to have approximately two newer samples of human motion for the update. The entire computation of the proactive path (including the computation of the probability cloud) took a mean time of 1.23 seconds. The resolution G (see Section III-B) was set equal to 25 intervals, while the path of the robot was composed by 43 waypoints.

Figure 7 reports some significant proactive paths computed for one of the experiments of the first group. In that experiment, the operator took all the parts from buffer 2 for the initial cycles, while switched to buffer 1 for the final ones (this explains the values for the probabilities P_{11} and P_{12} reported in Figure 7). As can be seen in Figure 7, when P_{11} is greater than P_{12}, the path is more deformed in its initial part (the one for which the robot passes close to the human when this latter is executing action h_1), see the picture on the right of Figure 7. On the opposite, when P_{11} is lower than P_{12}, the path is more deformed toward the end, see the picture on the left of Figure 7.

C. Results

The performance obtained with the two groups of participants is now discussed, from both an objective and a subjective point of view. In all the experiments, a total amount of 4 boxes were assembled. Figure 8 shows the distribution of the distance between the human and the robot during the experiments. There is a clear statistical evidence that it increases when applying proactive paths (single-tailed Wilcoxon rank sum test with confidence $\alpha = 0.05$ returns $r = 1.0 - 10^{-7}$). Figure 9 reports the percentage of time the robot speed went below the 1.0% of its nominal value. Also in this case the latter quantity is reduced when considering pro active paths (single-tailed Wilcoxon rank sum test with confidence $\alpha = 0.05$ returns $r = 0.9984$). Apart from the quantitative metrics previously reported, the subjects were also asked to fill in a survey, whose results are reported in Figure 6. It is interesting to note that subjects that worked with the robot performing pro-active paths, seem to indicate that the perceived fluency and safety of the interaction have increased.

VI. Conclusions

This work tackled the problem of proactive path planning for cobots. A probability description of the volume that the human will occupy during the robot's motion was considered

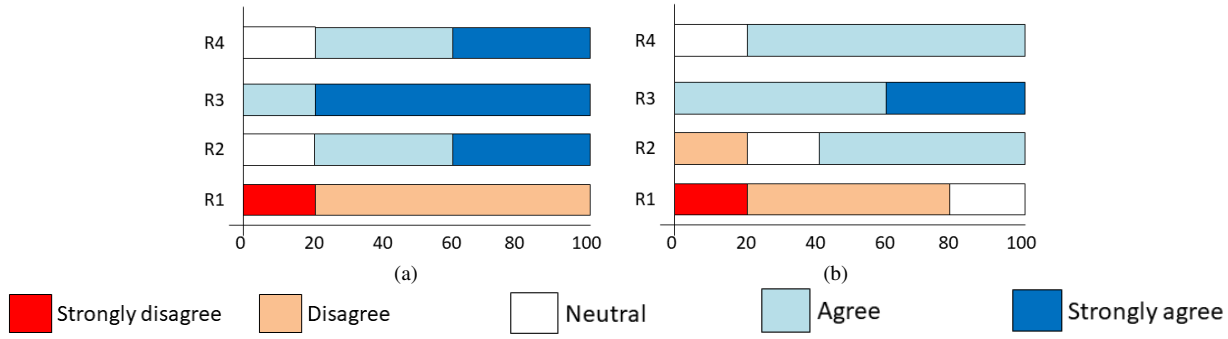

Fig. 6: Subjects rate in percentage to the following quotes. R1: "The robot movements were unnatural and strange"; R2: "The interaction with the robot was fluent"; R3: "The interaction with the robot was safe"; R4: "The robot was able to efficiently forecast the human trajectories". Picture on the left refers to the group of participants for which pro active paths were executed. Picture on the right, to the group for which nominal paths were executed.

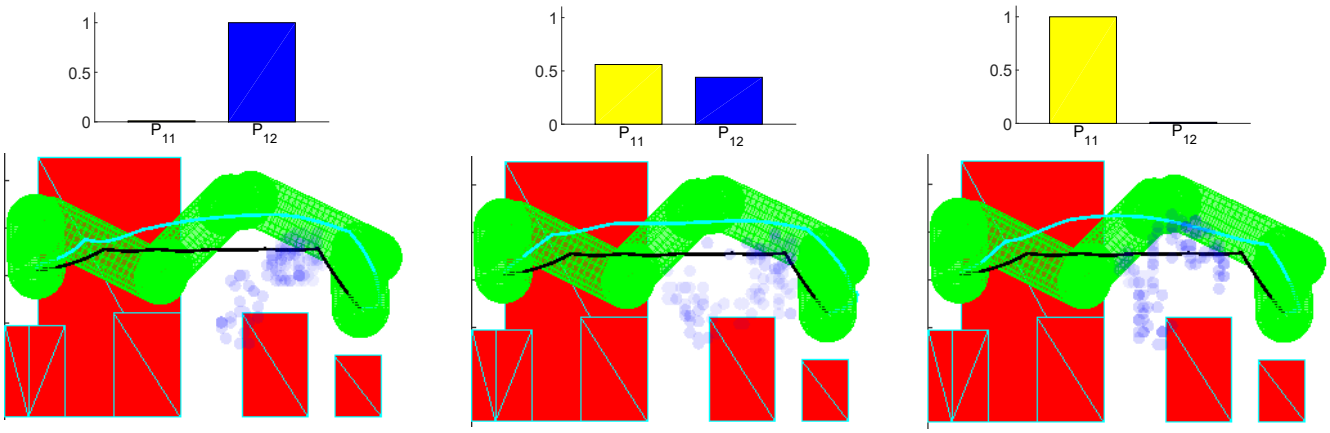

Fig. 7: Examples of proactive paths obtained for some invocations of STOMP (Red shapes are the fixed obstacles populating the scene) during the experiments. For every column: on the top, the probabilities considered for the computation of the probabilistic cloud; in the bottom, two distinct views of the proactive paths computed (cyan), compared with the initial nominal one (black). The probability clouds considered for planning are depicted as a series of blue points whose intensity is proportional to the probabilities contained in γ.

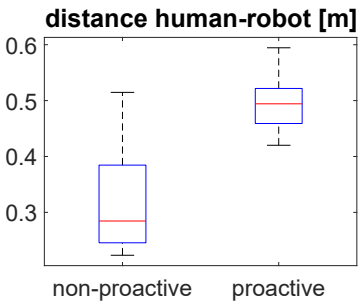

Fig. 8: Distribution of the distance between the human and the robot during the experiments.

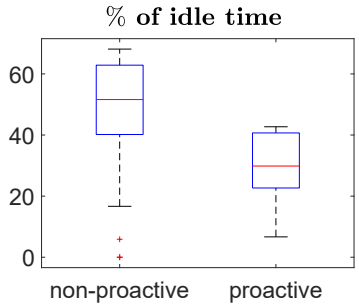

Fig. 9: Distribution of the percentage of idle time for the robot during the experiments.

for optimizing the robotic paths. The applicability of the proposed approach was demonstrated in a realistic collaborative interaction.

The adoption of proactive paths has improved the interaction, since the robot was able to dodge in advance the human, leading to a reduction of stress for the operator as well an improved productivity.

For future studies, it might be interesting to adapt the values of parameters $\lambda_{1,2}$ involved in the computation of the proactive paths, by considering an estimated level of confidence owned by the operator.

REFERENCES

[1] S. Robla-Gómez, V. M. Becerra, J. R. Llata, E. Gonzalez-Sarabia, C. Torre-Ferrero, and J. Perez-Oria, "Working together: a review on safe human-robot collaboration in industrial environments," *IEEE Access*, vol. 5, pp. 26 754–26 773, 2017.

[2] F. Flacco, T. Kröger, A. De Luca, and O. Khatib, "A depth space approach to human-robot collision avoidance," in *Robotics and Automation (ICRA), 2012 IEEE International Conference*.

[3] M. Ragaglia, A. M. Zanchettin, and P. Rocco, "Trajectory generation algorithm for safe human-robot collaboration based on multiple depth sensor measurements," *Mechatronics*, 2018.

[4] A. Casalino, P. Rocco, and M. Prandini, "Hybrid control of manipulators in human-robot coexistence scenarios," in *American Control Conference (ACC), 2018 IEEE International Conference*. IEEE.

[5] M. Awais and D. Henrich, "Human-robot collaboration by intention recognition using probabilistic state machines," in *Robotics in Alpe-Adria-Danube Region (RAAD), 2010 IEEE 19th International Workshop on*. IEEE, 2010, pp. 75–80.

[6] O. C. Schrempf, U. D. Hanebeck, A. J. Schmid, and H. Worn, "A novel approach to proactive human-robot cooperation," in *Robot and Human Interactive Communication, 2005. ROMAN 2005. IEEE International Workshop on*. IEEE, 2005, pp. 555–560.

[7] H. Khambhaita and R. Alami, "Viewing robot navigation in human environment as a cooperative activity," 2017.

[8] E. A. Sisbot, L. F. Marin-Urias, R. Alami, and T. Simeon, "A human aware mobile robot motion planner," *IEEE Transactions on Robotics*, vol. 23, no. 5, pp. 874–883, 2007.

[9] P. A. Lasota and J. A. Shah, "Analyzing the effects of human-aware motion planning on close-proximity human–robot collaboration," *Human factors*, vol. 57, no. 1, pp. 21–33, 2015.

[10] S. Pellegrinelli and N. Pedrocchi, "Estimation of robot execution time for close proximity human-robot collaboration," *Integrated Computer-Aided Engineering*, no. Preprint, pp. 1–16, 2017.

[11] J. M. Wang, D. J. Fleet, and A. Hertzmann, "Gaussian process dynamical models for human motion," *IEEE transactions on pattern analysis and machine intelligence*, vol. 30, no. 2, pp. 283–298, 2008.

[12] C. E. Rasmussen, "Gaussian processes in machine learning," in *Advanced lectures on machine learning*. Springer, 2004, pp. 63–71.

[13] D. J. Meagher, *Octree encoding: A new technique for the representation, manipulation and display of arbitrary 3-d objects by computer*. Electrical and Systems Engineering Department Rensseiaer Polytechnic Institute Image Processing Laboratory, 1980.

[14] G. Van Den Bergen, "Proximity queries and penetration depth computation on 3d game objects," in *Game developers conference*, vol. 170, 2001.

[15] M. Kalakrishnan, S. Chitta, E. Theodorou, P. Pastor, and S. Schaal, "Stomp: Stochastic trajectory optimization for motion planning," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*. IEEE, 2011, pp. 4569–4574.

[16] A. M. Zanchettin and P. Rocco, "Path-consistent safety in mixed human-robot collaborative manufacturing environments," in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*. IEEE, 2013, pp. 1131–1136.